

REMARKS

Claims 1-21 are pending in the above-identified application. Claims 1-21 were rejected in the Office Action dated August 24, 2004. No claims are amended in this Response.

Accordingly, claims 1-21 are at issue in the above-identified application.

I. 35 U.S.C. § 103(a) Obviousness Rejections of Claims

Claims 1-11 and 15-21 were rejected under 35 U.S.C. 103(a) as being unpatentable over *Hetherington et al.* (U.S. Patent No. 6,275,810) in view of *Kaufman* (U.S. Patent No. 5,313,647). In addition, claims 12-14 were rejected under 35 U.S. 103(a) as being unpatentable over *Kaufman* in view of *Hetherington et al.* Applicants respectfully traverse the rejection of the claims.

Applicants respectfully traverse the rejection to claims 1 and 12, for example, at least because the Examiner has not demonstrated a *prima facie* case of obviousness. In the Office Action dated August 24, 2004, the Examiner reiterates the assertion that “Hetherington does not explicitly teach the step of mapping so that it overlays virtual addresses of the master process.” However, the Examiner asserts that “Kaufman teaches that when spawning, the child would overlay the parent’s virtual memory address (vm_folk (sic) to duplicate a parent process’s virtual memory information for a child process, col. 31 line 14-20 and col. 2 lines 1-5).” (Office Action, p. 3).

Applicants respectfully traverse the rejection at least because, as disclosed in *Kaufman*, a child process is formed by copying but does not share or overlay the virtual addresses of the parent. Information of the parent may be copied in *Kaufman*, but the virtual address is not

overlaid or the same virtual address, for example. Copying or duplication of information is not the same as sharing the same virtual addresses, and *Kaufman* equates forking with copying or duplication. This is also reiterated by the Examiner in the Office Action (“duplicate... information for a child process”; Office Action, p. 3). For example, *Kaufman* states “[a] fork element can create a second process that initially duplicates a first one and can initiate generation of a new-process signal in connection with creation of the second process.” (Abstract; emphasis added). Nowhere does it teach or suggest that the processes share or overlay the same virtual addresses. In addition, *Kaufman* states “many computer systems, for example, those running under UNIX or UNIX-like operating systems, permit process duplication, or forking. Forking causes one process to replicate, spawning a new process. The first process, referred to as the parent, continues in the normal manner after the fork. The spawned process, or child, though initially identical to the parent, can be executed in a different manner.” (Col. 2, ll. 1-5; emphasis added) In the portion cited by the Examiner, *Kaufman* states “[t]he vm system executes the steps of the procedure vm_fork to duplicate a parent process's virtual memory information for a child process.” (Col. 31, ll. 14-21). As stated above, duplication of information is not sharing or overlaying the same virtual addresses.

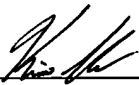
As a result, Applicants submit that claim 1 is patentable for at least the above-mentioned reasons. Claims 2-7 depend on claim 1 and are therefore patentable at least for the same reasons. Claims 8, 12 and 15 are patentable for at least the same reasons as claim 1. Furthermore, claims 9-11, 13-14, and 16-21 depend on claims 8, 12 and 15 respectively and are therefore patentable at least for the same reasons.

II. Conclusion

In view of the above remarks, Applicants submit that all claims are allowable over the cited prior art and respectfully request early and favorable notification to that effect.

Respectfully submitted,

Dated: November 24, 2004



Kimani P. Clark
Registration No. 45,969

Customer No. 26263
Sonnenschein Nath & Rosenthal LLP
P.O. Box 061080
Wacker Drive Station
Chicago, Illinois 60606-1080
Telephone: 202/408-9214
Facsimile: 312/876-7934